NaturalX Glossary

This glossary explains the terminology used in this documentation.

Activation Policies

An activation policy is an attribute of a NaturalX class which defines whether it runs within its own exclusive Natural session or whether it may share a Natural session with other classes.

NaturalX combines the different options supported by DCOM in the form of the following three activation policies:

- *ExternalMultiple*
- *ExternalSingle*
- *InternalMultiple*

The activation policy of a class can be set as part of the REGISTER command, in the DEFINE CLASS statement or with the profile parameter ACTPOLICY=*activation-policy* (for OS/390, DCOM=(ACTPOL=*activation-policy*)).

AppID

In the system registry, each application is represented by an AppID. The AppID is a globally unique ID which can be found under the registry key HKEY_CLASSES_ROOT\AppID. DCOM uses the AppID to group classes to applications. Also, for example, security settings are defined on the basis of the AppID. Natural creates for each server ID one AppID in the registry.

For further information, see the section Server ID.

Class

Following the object-based programming approach, NaturalX classes encapsulate data structures (objects) with corresponding functionality (methods).

The internal structure of the objects of a class object is defined with a data area (object data area). The methods of a class are implemented as subprograms.

NaturalX classes can be made known to DCOM using the Natural command REGISTER, after which they are accessible in a network.

Classes can be *internal*, *external*, or *local*.

For further information on classes, see the sections Programming Techniques, Defining Classes, Using Classes and Objects, and Internal, External and Local Classes.

Class GUID

If a Natural class is to be registered as a DCOM class, a globally unique ID (GUID) must be defined for the class, to make sure it can be unambiguously identified in a network. In Natural, a GUID is assigned to a class in the ID clause of the DEFINE CLASS statement. A GUID is represented by an alphanumeric constant which can be generated in the data area editor.

Copyright Software AG 2001

Class Name

The name defined in the *class-name* operand in the DEFINE CLASS statement. This name is used in the CREATE OBJECT statement to create objects of that class.

Class Module Name

The name of the Natural module in which a Natural class is defined.

COM

Component Object Model. Microsoft specification for binary component API. Standardizes communication between components on a binary level. This component software model specifies how software components interact, irrespective of the programming language in which they were implemented.

For further information, see the Microsoft COM specification.

COM Class

A class that makes its methods and properties available to clients through interfaces that comply with the COM specification.

For further information, see the section COM.

DCOM

Distributed Component Object Model. DCOM extends COM to a distributed component software model which specifies how software components interact in a distributed environment.

With EntireX DCOM, Software AG has also made the DCOM technology available on UNIX and mainframe platforms.

 Copyright Software AG 2001

External Class

A Natural class can be a local, an internal or an external class. This depends on the way the class was registered. An external class is a class that has been registered as a DCOM class with the REGISTER command option *ExternalSingle (ES)* or *ExternalMultiple (EM)*. Objects of external classes can be created and accessed by other processes. (With Natural on Windows 98/NT/2000 and UNIX, objects of external classes can also be created in the client process, provided the client process is at the same time a server process for the class.)

For further information, see the sections Local Class and Internal Class.

GUID

A GUID (globally unique identifier) is a constant that is guaranteed to be unique worldwide in the COM/DCOM model. It is used to unambiguously identify classes and their interfaces in a network. If a Natural class is to be registered as DCOM class, a GUID must be assigned to the class and to each of its interfaces. In Natural, a GUID is represented by an alphanumeric constant which can be generated in the data area editor.

For further information, see the section Globally Unique Identifiers (GUIDs).

HFS

Hierarchical File System. - UNIX file system available with OS/390 UNIX Services.

Instance

In the object-oriented programming model, data structures and functions (so called *methods*) are packaged together in objects. Each object belongs to a class, which describes the internal structure of the object and its interfaces, properties and methods. If an object belongs to a certain class, it is also called an instance of that class.

Interface

Interfaces are used by classes to provide clients with services. An interface is a collection of methods and properties. A client accesses these services by creating an object of the class and using the methods and properties of its interfaces.

You define an interface as follows:

- Define the INTERFACE clause to specify an interface name.
- Define the properties of the interface with PROPERTY definitions.
- Define the methods of the interface with METHOD definitions.

Copyright Software AG 2001

Interface GUID

If a Natural class is to be registered as a DCOM class, a globally unique ID (GUID) must be defined for each of its interfaces, to make sure the interfaces can be unambiguously identified in a network. The GUID is assigned to an interface in the ID clause of the INTERFACE statement. In Natural, a GUID is represented by an alphanumeric constant, which can be generated in the Data Area Editor.

Interface Inheritance

Interface inheritance means giving different classes the same interfaces, but implementing the interfaces differently in the different classes. This makes it possible to write client programs that only rely on these interfaces and are able to work with any class that has these interfaces.

Internal Class

A Natural class can be a local, an internal or an external class. This depends on the way the class was registered. An internal class is a class that has been registered as a DCOM class with the REGISTER command option *InternalMultiple (IM)*. Objects of internal classes can not be created by other processes, but they can be accessed by other processes. This requires that the object has been passed to the client process for example as return value of a method.

For further information, see the sections Local Class and External Class.

Local Class

A Natural class can be a local, an internal or an external class. This depends on the way the class was registered. A local class is a class that has not been registered as a DCOM class. Therefore, objects of local classes can neither be created nor accessed by other processes, but only by programs in the current Natural session.

For further information, see the sections Internal Class and External Class.

Method

A method is a function that an object/instance of a class can perform when requested by a client.

NaturalX Client

A NaturalX client is a process which creates or accesses NaturalX objects.

 Copyright Software AG 2001

NaturalX Server

A NaturalX server is a process which manages one (Windows 98/NT/2000 and UNIX) or multiple (OS/390) Natural sessions. The Natural sessions managed by a NaturalX server are used to host COM objects.

Natural Session

A Natural session is the user-dependent Natural runtime context required for the Natural runtime system to execute Natural programs.

Object

In the object-oriented programming model, data structures and functions (so-called *methods*) are packaged together in objects. Each object belongs to a class, which describes the internal structure of the object and its interfaces, properties and methods.

Object Data Area - ODA

The object data area is the place where the current values of all properties of an object are stored. Also other variables can be defined in the object data area, which are not accessible by clients as properties, but just used by the methods of the object to maintain an internal state of the object. The structure of the object data area of all objects of one class is specified in the OBJECT USING clause in the DEFINE CLASS statement. An object data area is nothing else than a local data area, and in fact it is also created in the data area editor as a local data area.

Object Data Variable

Each property needs a variable in the object data area of the class to store its value - this is referred to as the object data variable.

ProgID

The ProgID (programmatic identifier) of a DCOM class is a meaningful name by which the class is identified in client programs. For Natural classes, the name defined in the *class-name* operand in the DEFINE CLASS statement is written into the registry as a ProgID when the class is registered as a DCOM class with the REGISTER command.

Property

Properties are attributes of an object that can be accessed by clients. In Natural classes, property values of an object are stored in the object data area. Therefore an object data variable must be assigned to each property.

For further information, see the section Object Data Variable.

Copyright Software AG 2001

Registry

A repository containing operating system information. The registry also contains information about DCOM classes and their assignment to servers.

Registry Key

Registry keys are entries made in the system registry of the server when a class is registered. Registry keys are also added in the client system registry when the client registration file is executed.

For detailed background information about the registry keys and their administration, please refer to the registry documentation of the appropriate platform.

Server ID

A server ID is a character string that identifies a NaturalX server. The server ID is a Natural-owned key in the system registry, keeping together all classes that belong to a given NaturalX server. It is an arbitrary alphanumeric string of 32 characters which does not contain blanks and which is not case sensitive. The server ID is defined with the Natural parameter COMSERVERID=*serverid* (for OS/390, DCOM=(SERVID=*serverid*)).

Type Information

When a Natural class is registered as a DCOM class, a type library is generated for the class and connected to the class by a registry entry. Clients can use the type information contained in the type library to check the descriptions of the interfaces, methods and properties of the class either at compile time or at runtime.

Type Library

When a Natural class is registered as a DCOM class, a type library is generated for the class and connected to the class by a registry entry. Clients can use the type information contained in the type library to check the descriptions of the interfaces, methods and properties of the class either at compile time or at runtime.

 Copyright Software AG 2001